Evolution of the Metadata in the Ontology-based Knowledge Management Systems

Ljiljana Stojanovic[1], Nenad Stojanovic[2], Siegfried Handschuh[2]
[1]FZI Research Center for Information Technologies
at the University of Karlsruhe, Germany
[2]Institute AIFB - University of Karlsruhe, Germany

Abstract: An ontology-based knowledge management system uses an ontology to represent explicit specification of a business domain and to serve as a backbone for providing and searching for knowledge sources. But, dynamically changing business environment implies changes in the conceptualisation of a business domain that are reflected on the underlying domain ontologies. Consequently, these changes have effects on the performances and validity of the KM system.
In this paper we present an approach for enabling consistency of the description of knowledge sources in an ontology-based KM system in the case of changes in the domain ontology. This approach is based on our research in the ontology evolution and ontology-based annotation of documents. The proposed method is implemented in our semantic annotation framework so that efficient acquiring and maintaining of ontology-based metadata is supported.

1. Introduction

In the dynamically changing world of business, the competitiveness of companies depends heavily on the possibility to find, for a given problem, the right knowledge in the right moment. This view presumes existence of knowledge sources and focuses on the acquiring, using and validating knowledge sources – so called supply-side of the knowledge management system [Mc01]. Practically, such KM approach is based on the knowledge integration process [Fi01], in which heterogeneous forms of knowledge sources (text, audio, video) should be integrated through unified searching interface in order to find right solution for a given problem. Expansion in the using Internet technologies for corporate IS implies using web portals as unique interface for providing and accessing content of various knowledge sources. The prerequisite for this integration is the unified description of the content of the knowledge sources – unified format and used vocabulary. As a promising integration structure appears ontologies that provide real-world and formal semantic of a domain theory. There are several approaches for using ontologies in KM [Fe00]. Ontology-based KM systems use ontologies as a backbone for providing and accessing knowledge sources. An ontology offers a terminology for the knowledge indexing and searching process. The main advantage comparing to key-word based indexing/searching is that an ontology is a formalised, common and shared description of a domain. Therefore, it provides a set of assumptions about intending meaning of used terms, e.g. when one searches for a knowledge source that is about the animal "jaguar" then it is avoided to retrieve a source that describes the "jaguar" car.

An ontology in a KM system is related to the business strategy and also indirectly to the business environment. Consequently in a fast changing environment it is obvious that an ontology as a domain backbone is also a matter of change. The changes have to be propagated to all descriptions, e.g. annotations of the knowledge sources in order to enable consistency. Although this change propagation problem has great impact on knowledge searching process, this problem is not well addressed in the KM literature [Ha00].

In this paper we present an approach that enable consistency in the annotations of knowledge sources in the case of changes in the domain ontology. The approach is based on our research in the area of ontology evolution and ontology-based annotation of documents [Ha01]. The proposed method is included in our semantic annotation framework CREAM so that efficient acquiring and maintaining of ontology-based metadata is supported.

The benefits of the proposed approach are manifold:

- In the case of changes in the domain ontology, annotations of knowledge sources can be automatically updated;
- An ontology-evolution model enables the categorisation of required/derived changes so that incorrectnesses which lead to the more critical decreasing of the system's performances can be managed firstly;
- A special ontology for maintenance of the annotation is introduced -- herein after called maintenance ontology. It offers new search possibilities for knowledge sources, not only according to the content, but also according to author, date, format, relevance and their combinations.

From the knowledge management system point of view the proposed approach will enable us to develop a robust knowledge management solution that copes with the high-changeable business conditions.

Paper is organised as follows: Section 2 describes the typical problems of the knowledge-management systems in the dynamically changing world of business. Section 3 explores the problem of changes in the ontology and analyses the effect of the change on the ontology itself and on the underlying objects. In section 4 we describe a method to analyse and propagation changes made in the ontology. Further we present an integration tool for implementing this method in a KM scenario. Before we conclude, we give a survey of related work in the categories knowledge management, ontology evolution and annotation environments.

2. Maintenance problem in an ontology-based Knowledge Management system

The frequency and variety of doing a business implies the production of tons of information in various representation formats (text, audio, video) and various levels of structures (structured, semi-, un-structured). Information are spread all overall a company: in business documents, technical documentations, manuals, legacy databases, and e-mails. Most of these items could be treated as very valuable knowledge sources for a particular problem. One of the most important tasks of a knowledge management system is to find effectively the appropriate content in all of these heterogeneous sources. All the systems of the first generation of KM [Mc01] are supply-side, which means that they are focused mainly on searching for relevant knowledge sources (knowledge integration process) and less (or not at all) on the knowledge production process [Fi01].

Using Internet infrastructure enhances in many ways "searching for knowledge" practices of an KM system [OL98], while the Web provides a possibility to integrate all of these sources on the presentation level: all of them could be presented to the user through a single interface – the Web browser. Therefore, a human expert could use the same Web interface (for example, searching engines like AltaVista.) to find relevant information stored in the text, pictures, video files. But, real experience teaches us that such an expert should not be only a domain expert (for example bio-chemist) but a searching-expert (for particular searching engine) as well.

The mentioned searching-problem lies in the structure of the current Web – it is designed only for human consumption - machines so far mainly help in better presenting of information, but with limited possibilities to process the content of the presented information. Therefore, the real integration of the knowledge sources has to be done by the formal introduction of an intermediate level (between syntax and presentation levels) that will help software agents to understand the content of the knowledge sources.

This machine understanding assumes [Fe00]:

- a formal understanding that allows the processing of the semantic by a computer;
- a real-world understanding that allows relating semantic of information to the common-shared meaning of humans.

It can be realised by annotating each knowledge source with a formal description of the content.

As a promising structure for realising such a machine understanding appears ontologies, an explicit specification of the conceptualisation of the domain of interest [Gr93]. Ontologies typically consist of definitions of concepts relevant for the domain, their relations, and axioms about these concepts and relationships. They provide a suitable format and a common-shared terminology for the description of the content of knowledge sources. In other words, each knowledge source should be semantic annotated, i.e. enriched with a metadata description [Ha01].

The semantic annotation resolves one of the common problems in the underlying ontology-based KM systems: the prediction game between indexers and users. An indexer attempts to predict which concepts a user will employ when searching for a particular knowledge source. In formulating a query, the user attempts to predict which index concepts are attached to the knowledge source he or she seeks. By using a given domain ontology one can annotate content of provided knowledge source in such a way that a knowledge-seeker can find that knowledge source easily, independently of its representation format – which is the vision of an ontology-based supply-side KM system [St01].

The suggested changes in the machine-understandable description of the content of the knowledge sources in the Web require changes in the basic WWW infrastructure, which leads to the second generation of the Web – so called Semantic Web [BL00]. The basic infrastructure for the Semantic Web is on the way and the presented KM scenario could be one of the "killer applications" for the Semantic Web.

However, there are several crucial problems, which should be resolved in order to realise this "KM dream". We mention only a few: how to define an ontology, how to support semantic annotation of the heterogeneous (audio, video) knowledge sources, how to define a query language on the metadata level. But from the point of view of a KM system the most important question arises: is the metadata assigned to a knowledge source valid, i.e. up-to-date?

As mentioned in the introduction, each KM system should reflect indirectly, implied by changes in the ontology, all the changes made in the business environment. Particularly, each change in the business conceptualisation (changes in the strategy of the company, in the market planning, in the customer segmentation) requires changes in the domain modelling and each change in the domain modelling should be reflected in the changes of the metadata description of the content of the knowledge sources. That does not mean that every change in business environment implies a change in the validity of knowledge sources - knowledge sources are valid but the view on the content of these knowledge sources is changed. From the annotation point of view the terminology used for the description of the content of the knowledge sources should be changed according to the business changes.

A knowledge management system could decrease its efficiency drastically in the case that some of the knowledge sources are annotated with an "old" ontology and that a "revised" ontology is used for searching. For a given query it would not only miss some relevant knowledge sources, but also deliver WRONG answers, for example in the case of the query: "Give me all knowledge sources that describe bonuses of our customers" given in figure 2.1.

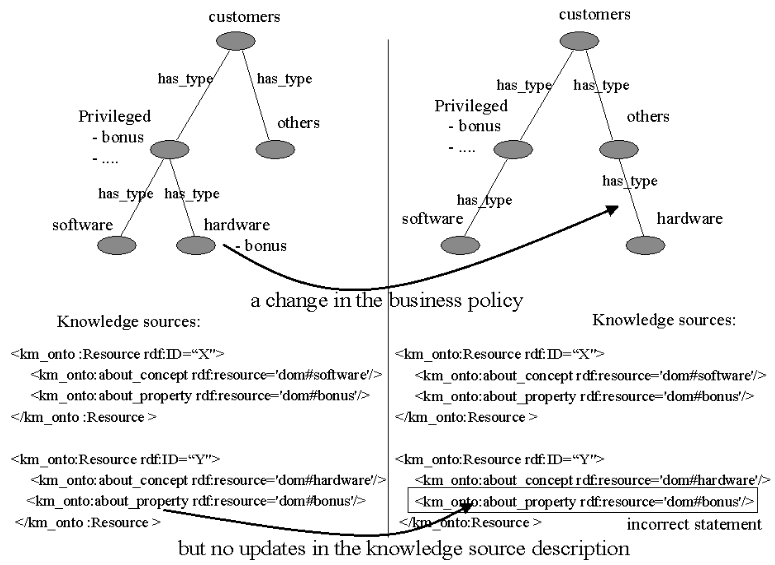

Figure 2.1. Incorrect statement – part of business ontology: Concept "customer" is divided in two subconcepts: "Privileged" and "others". "Privileged" customers are divided into "software" (only software is sold to them) and "hardware" (only hardware is sold to them). In the "old" ontology (left side) concepts "software" and "hardware" inherit a property "bonus" from their parent. In the "new" ontology (right side) the concept "hardware" changed its parent. However, the new parent does not contain the property "bonus". As a consequence, the knowledge source "Y", that is about bonus for customer who buy hardware, is incorrect while in the "new" ontology (new business policy) this type of customer has no bonus privilege. The meaning of used representation formalism is detailed in the section 3

In the next sections we describe a method (section 3, 4) to analyse and propagation changes made in the ontology. Further we present an integration tool (section 4) for implementing this method in a KM scenario.

3. Ontology evolution

One critical point in applying ontologies to real-world problems is that domains are changing fast (new concepts evolve, concepts change their meaning, new business rules are defined, etc.) and user needs are changing, too. Thus, the corresponding ontologies have to evolve as well. Ontology evolution is the timely adaptation of the ontology to the changed business requirements, to the trends in the ontological instances and to the way of using of the ontology-based applications, as well as the consistent management/propagation of these changes because a modification in one part of the ontology may generate subtle inconsistencies in other parts of the same ontology, in the ontology-based instances, depending ontologies and applications. This variety of causes and consequences of the ontology changes makes ontology evolution a very complex process (figure 3.1) that is described in the following.

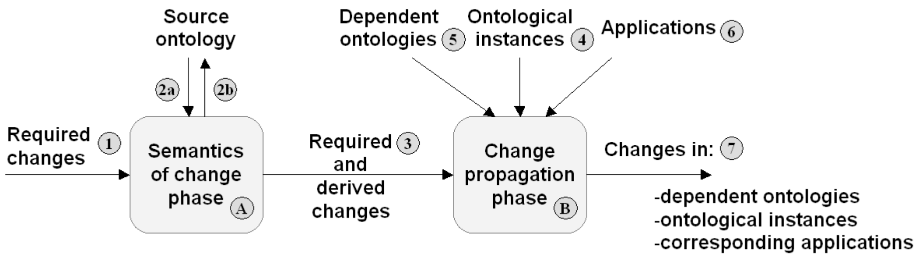

Figure 3.1. Phases in the ontology evolution process

Elementary changes in the ontology are shown in the table 3.1, including the addition and the deletion of all ontological entities. Modification (update) of any ontological entity is realized using the deletion of the old entity and the addition of the new entity. The single exception is the modification in the concept hierarchy because of the relation's inheritance. Change in the name of the entity is not considered, because every entity has unique identifier that is independent of the entity name.

Table 3.1. Elementary changes in the ontology

	Elementary change
Add	AddConcept, AddRelation, AddIsA, AddAxiom, AddDomain, AddRange
Delete	DeleteConcept,DeleteRelation,DeleteIsA,DeleteAxiom,DeleteDomain, DeleteRange
Modify	Modify_IsA

3.1 Semantics of change

An ontology has to be consistent according to its structure (concepts, inheritance graph, relations, axioms). This is "semantics of change" phase (cf. figure 3.1 (A)) that refers to the effect of the change on the ontology itself. In order to retain consistency of the ontology, set of required changes is expanded with the additional (derived) changes in ontology. For example, the deletion of relation domain can provoke the

deletion of the relation as well in the case that there are no other concepts defined as domain of this relation.
The additional changes in the ontology are derived automatically. The approach is based on the sound and complete set of axioms (provided with an inference mechanism) that formalises the dynamic of the ontology evolution. The compliance of the available ontology changes with the axioms automatically ensures ontology consistency, without need for explicit checking as incorrect ontology version cannot actually be generated [Fr00]. While the focus of the paper is on the knowledge management, we will omit here the description of our approach used for "semantics of change" and concentrate on the "change propagation" problem, which has great impact on the knowledge searching process.
Inputs of this phase (A) are required changes (1) and source ontology (2a) and outputs are list of required and derived changes (3) and modified source ontology (2b).

3.2 Change propagation

Potentially, an ontology change might corrupt the instances, dependent ontologies as well as application programs running against the ontology and/or the knowledge base. The task of the change propagation phase is automatically bringing all dependent elements to a consistent state after an ontology update has been performed. Block (B) in the figure 3.1 depicts this phase. Output is the list of changes (7), which have to be done. In the rest of the section we will analyse the effect of the change propagation on the corresponding inputs.

Effect of changes on the dependent ontologies
An ontology update might corrupt ontologies that depend on the modified ontology. They are built from the modified ontology or they import it. This problem could be solved by recursive applying ontology change procedure on these ontologies in order to preserve their conceptual, structural and behavioural consistency [Fr00].

Effect of changes on the ontological instances
When the ontology is modified, the instances need to be changed in such a way that the ontology and instances remain consistent with each other. Basically, if the ontology is modified instances must be transformed to confirm to the modified ontology. It means that continuous adaptation of the annotated information to the new semantic terminology and relationships is necessarily.

Effect of changes on the applications
Changes in the ontology might invalidate applications that are already running on top of the ontology and the knowledge bases, especially if they rely on certain schema characteristics, which are lost after the ontology update. In the ideal case, the conceptual knowledge that is necessary for an application should be merely specified in the ontology. However, practice applications also use an internal model that may become incompatible with the ontology [KF01]. Moreover, although the application's programs are written to be as generic as possible, there are a certain number of "hard-coded" elements that should be treated special in some way. In most of the web applications, where some queries are "hard-coded" into the service that is invoked as a response on the specific action, the query rewriting process is needed [FL96].

4. Evolution of the metadata

This section introduces the backbone of our approach - evolution ontology that supports, alleviates and automates the evolution process. Thereupon, we present our method for solving the change propagation problem and our annotation framework, which integrates the ontology evolution process.

4.1 Evolution ontology

Since ontology evolution requires additional meta-level reasoning capabilities that allow inspecting changes and their logical dependencies, we define a special ontology, so-called evolution ontology. We distinguish between domain ontology that is changed and the evolution ontology that enables better management of these changes.
Ontological changes are represented using the top level concept "*Change*", its subconcepts (AddConcept, AddRelation, etc.) and its relations [Ol99]. For every change, it is also useful to know who is *author* of the change and when it is happened (*time*). The *cause* of the change is used to represent the source of the change (business requirements or the learning process) and the *relevance* of the change describes whether and how it can fulfil the requirements. Also, ontology evolution is managerial process and it needs some properties to support decision-making like *cost*, *priority*, etc. *Order* of the changes is also very important while it enables recovery of implemented changes. Moreover, change propagation cannot be done after every change in the ontology (it requires too much time) even thought the change causes instance inconsistency. Consequently, only the order of the changes can guaranty that the instances "picture" the real status of the ontology structure. To solve semantics of change problem, the evolution ontology contains axioms that derive additional changes. Similarly to the ordering of the change: this type of the dependency between changes is represented as a relation *parentChange*.
The second part of the evolution ontology represents semantic information about the domain ontology explicitly (relations *prototypical, primary_key*, etc.), because the conceptual structure of the evolution ontology aims to provide enough mechanisms to deal with problems of syntax as well as semantic inconsistencies that arise when the domain ontology is changed [TB01]. The third part of the evolution ontology aims to support data-driven self-improvement of the domain ontology. For example, the fact that these are no instances of some concept is a sign that this concept should be deleted. We enforce formal discovering of changes by representing these heuristics as axioms in the evolution ontology.
The benefits of using the evolution ontology are manifold: First, changes are formally represented. Second, a history of changes is stored. Third, based on the formal representation and the history of changes the change-propagation problem may be approached. Using the same representation model for the ontology and analysis of changes simplifies storage and allows reuse of system components like searching.

4.2 Evolution of the metadata

In this section we present our method for the change propagation problem based on consistency analysis of already existing metadata and the performed change in the domain ontology. It is divided into three steps described in the following.

Metadata capturing

When an ontology is modified, instances need to be changed in such a way that the ontology and instances remain consistent with each other. If the instances are on the Web, they are collected in the knowledge base using tools like focused crawler[1] (process "capture" in the figure 4.1). In order to speed up the whole change propagation process, only the instances that depend on the change are gathered. This dependency information is obtained from the instance of the evolution ontology that represents the performed change. Moreover, the output of this step is one list that makes references between located instances and Web documents.
The main problem is how to find an application that uses the ontology that is changed. An application can be semi-automatic maintained only if exists metadata describing which ontology and/or ontological entities that application uses. Thus, annotation of applications is necessary.

Metadata analysis

In the second step, automatic translation of the instances is performed according to the changes in the ontology [SSV02]. In order to avoid overhead of the system, which may heavily increase if the changes are performed every time the ontology has to be modified, the categorisation of the changes is embedded in the evolution ontology. We distinguish between:

- ontology-extending changes that do never have an impact on the existing instances (e.g. creating a new relation);
- changes that provoke syntax inconsistencies in the ontological instances (e.g. deleting a concept that already has instances);
- changes that provoke semantic inconsistencies in the ontological instances (e.g. creating a new sibling concept does not lead to the invalidity in the set of instances but an analysis of the meaning of the instances is needed).

The axiomatic part of the evolution ontology enables the verification of the formal characteristic of the instances. The analysis of the semantic consistency is based on the meta information (e.g. primary_key) defined in the evolution ontology.
This step provides an output in the form of list of modified instances with the reference to the corresponding resource (knowledge source). Only this step is performed in the case that instances are already gathered in the knowledge base.

Generation of a proposal for modifications

In the last step "out of date" instances on the Web are replaced with the corresponding "up-to-date" instances. As depicted in the figure 4.1, some modifications of the instances can be done automatically (process "update"), but for the instances that are "write-protected" the notification has to be sent (process "notification) to the author of the annotation in order to inform her/him about the changes and to suggest how to correct the instance. Information about author is saved in the property "Author" in the evolution ontology.
Using the method for metadata evolution does not solve all problems. However, we provide guidelines, which suggest which resources' metadata have to be checked, and eventually changes to run again the changed ontology.

[1] Kaon.semanticweb.org

4.3 Framework

In order to support the proposed approach for ontology evolution based on the maintenance of the instances we have adapted our CREAM framework [Ha01] presented in the figure 4.1. The Evolution Ontology, Evolution Component and related links are the new elements and they are described in the previous section.

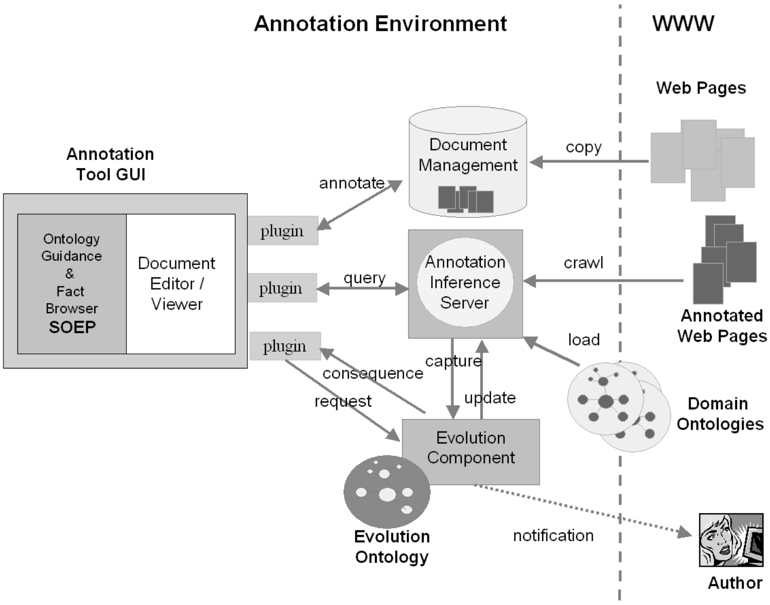

Figure 4.1. Architecture of CREAM

Document Editor/Viewer: The document editor/viewer visualizes the document content and the annotations.

SOEP[2] - Ontology and Fact Editor: The user can browse and edit the ontology and retrieve for one concept all instances or for one instance all properties.

Crawler: The creation of relational metadata must take place within the Semantic Web. During metadata creation subjects must be aware of which entities exist already in their part of the Semantic Web. This is only possible if a crawler makes relevant entities immediately available.

Annotation Inference Server: Relational metadata, proper reference and avoidance of redundant annotation require querying for instances, i.e. querying whether and which instances exist. For this purpose as well as for checking of consistency, we provide an annotation inference server in our framework. The annotation inference server reasons on crawled and newly created instances and on the ontology. It also serves the ontological guidance and fact browser, because it allows querying for existing concepts, instances properties.

Document Management: In order to avoid redundancy of metadata creation efforts, it is not sufficient to ask whether instances exist at the annotation inference server. When a metadata creator decides to capture knowledge from a Web page, he does not

[2] kaon.semanticweb.org

want to query for all single instances that he considers relevant on this page, but he wants information, whether and how this Web page has been annotated before. Considering the dynamics of HTML pages on the web, it is desirable to store annotated web pages together with their annotations. When the web page changes, the old annotations may still be valid or they may become invalid. The metadata creator must decide based on the old annotations and based on the changes of the web page.

5. Related Work

Knowledge management and annotation/ontology evolution

As known to authors the problem of maintaining description (annotations) of knowledge sources in an ontology-based KM system in the case of changes in the domain ontology is not treated in the literature and therefore we here present an analyse of the annotation systems for the knowledge management purposes. The last presented system gives the best view on the maintenance problem in the knowledge management community. Annotate [Gi99] is a system that use information retrieval methods to support KM in an organisation. It enables document annotations on the web and captures global usage history. Annotate is not ontology-based and therefore does not treat the problem of managing validity such knowledge item descriptions. In [DPP00] paper author presents several issues with the design and implementation of organisation memories in distributed companies. They have designed a tool, based on the domain model in the form of ontology, capable to capture the content of the documents and the context, in which they were created. A sophisticated retrieval engine can retrieve the annotated documents based on their context. The presented system seems very similar to ours; it has very suitable user interface which support process of creating document annotations, it is integrated in the general ontology engineering environment, but it is not adapted to new web infrastructure (Semantic Web) and does not consider ontology evolution problem.

A very interesting, field research study of managing changes in a knowledge management system is given in [Ha00]. The authors consider two types of changes: (i) functional changes that are about new KM-systems in the organization, new versions of a KM-system and new features in one KM-system and (ii) structural changes that deal with new business models, new subsidiaries and new competencies in the organisation. The results of the study show that managing the evolution of KM-systems on an ad hoc basis can lead to unnecessary complexity and KM-systems failures and that KM research has paid little attention to the evolution of KM-systems.

Ontology evolution

There are very few approaches investigating the problems of changing in the ontologies. The most similar approach to our approach is described in the paper [KF01]. As the authors also mentioned the most important flaw is the lack of a detailed analysis of the effect of specific changes on the interpretation of data.

The problem of schema evolution and schema versioning support has been extensively studied in relational and database papers. [Ro96] provides an excellent survey on the main issues concerned. [Fr00] introduces an approach to schema versioning, which considers a (conceptual) schema change as a (logical) schema augmentation. In contrast to our approach, this semantic approach does not address the change propagation problem, which concerns the effects of schema changes on the underlying instances. For the change propagation problem, several solutions have been proposed and implemented in real systems. In all cases, simple default mechanisms can be used or user-supplied conversion functions must be defined for non-trivial extant object updates. However, there are no approaches that treat data on the web.

Annotation

We know of three major systems that intensively use knowledge markup in the Semantic Web, viz. SHOE [HH00], Ontobroker [De99] and WebKB [PP99]. All three of them rely on knowledge in HTML pages. They all started with providing manual mark-up by editors. However, our experiences [Er00] have shown that text-editing knowledge mark-up yields extremely poor results, viz. syntactic mistakes, improper references, and all the problems sketched in the scenario section. The approaches from this line of research that are closest to CREAM are the SHOE Knowledge Annotator and the WebKB annotation tool. The SHOE Knowledge Annotator is a Java program that allows users to mark-up webpages with the SHOE ontology. The SHOE system [Lu97] defines additional tags that can be embedded in the body of HTML pages. The SHOE Knowledge Annotator is rather a little helper (like our earlier OntoPad [Fe99], [De99]) than a full-fledged annotation environment. WebKB uses conceptual graphs for representing the semantic content of Web documents. It embeds conceptual graph statements into HTML pages. Essentially they offer a Web-based template like interface like knowledge acquisition frameworks described next.

6. Conclusion

Ontology used in an ontology-based KM system is related to the business strategy and indirectly to the business environment. In the highly changed environment it is obvious that an ontology as a domain backbone is also a matter of change. The changes in the ontology have to be propagated to all ontology-based descriptions of the knowledge sources in order to enable consistency of the searching process.

In this paper we have presented an approach for enabling consistency of the descriptions of the knowledge sources in the case of the changes in the domain ontology. The approach is based on our research in the ontology evolution and the ontology-based annotation of the Web documents. The proposed method is implemented in our semantic annotation framework so that efficient acquiring and maintaining of the ontology-based metadata is supported.

The proposed approach has many benefits: automatic updating of the ontology-based description of the content of the knowledge sources, new possibilities for searching for the knowledge sources according to the author, date, format of the knowledge sources, to name but a few. From the knowledge management system point of view the proposed approach enables us to develop robust knowledge management solution, which copes with the high-changeable business conditions.

Combining this approach with the ontology learning methods, which enable learning ontologies from the knowledge sources, leads us in the some kind of self-organising knowledge management systems.

Acknowledgements: We thank our colleagues and students at the Institute AIFB, University of Karlsruhe, and FZI Research Center for Information Technologies at the University Karlsruhe, without whom the research would not have been possible – especially Steffen Staab and Alexander Mädche. Research for this paper was partially financed by EU in the IST-2000-28293 project "OntoLogging" and by US Air Force in the DARPA-DAML project "OntoAgent".

7. Literature

[BL00] Berners-Lee, T.: XML 2000 – Semantic Web talk, 2000 http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

[De99] Decker, S.; Erdmann, M.; Fensel, D. and Studer, R.: Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In Database Semantics: Semantic Issues in Multimedia Systems, pages 351–369. Kluwer Academic Publisher, 1999.

[DPP00] Dzbor, M.; Paralic, J.; Paralic., M.: Knowledge Management in a Distributed Organisation. In Advances in Networked Enterprises, Berlin, Germany, September 2000.

[Er00] Erdmann, M; Maedche, A.; Schnurr, H.-P. and Staab, S.; From manual to semi-automatic semantic annotation. Proc. COLING 2000, Luxembourg, August 2000.

[Fe00] D.Fensel The role of Ontologies in Information Interchange. Proc. 2nd International Scientific and Practical Conference on Programming UkrPROG'2000, Ukraine, 2000.

[Fe00a] Fensel, D.; Harmelen F; Klein, M; Akkermans, H; Broekstra, J.: On-To-Knowledge: Ontology-based Tools for Knowledge Management: eBusiness and eWork, 2000

[Fe99] Fensel, D.; Angele, J.; Decker, S.; Erdmann, M.; Schnurr, H.-P.; Staab, S.; Studer, R. and Witt, A.: On2broker: Semantic-based access to information sources at the www. In Proc. of the World Conference on the WWW and Internet (WebNet 99), Honolulu, Hawaii, USA, 1999.

[Fi01] Firestone, J.: Knowledge Management Process Knowledge Management Process Methodology: An Overview, Knowledge and innovation: Jornal of the KMCI, Volume 1, 2001

[FL96] Ferrandina, F.; Lautemann S.E.: An integrated approach to schema evolution for object databases, OOIS 1996: 280-294

[Fr00] Franconi, E.; Grandi, F.; Mandreoli, F.: A semantic approach for schema evolution and versioning in object-oriented databases. Proc. CL2000, 2000

[Gi99] Ginsburg, M.; Kambil, A.: Annotate: A Web-based Knowledge Management Support System for Document Collections, Proc. of HICSS-32, 1999.

[Gr93] Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition, 5:199–220, 1993

[KF01] Klein, M.; and Fensel, D.: Ontology versioning for the Semantic Web, Proc. International Semantic Web Working Symposium (SWWS), USA, July 30 - August 1, 2001.

[Ha01] Handschuh, S.; Staab, S.; Mädche: A. CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. ACM, K-CAP 2001, October, 2001

[Ha00] Hardless, C.; Lindgren, R.; Nulden, U. and Pessi, K.: The Evolution of knowledge management system need to be managed, http://www.viktoria.informatik.gu.se/groups/ KnowledgeManagement/ Documents/kmman.pdf, 2000

[HH00] Heflin, J. and Hendler, J.: Searching the web with shoe. In Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01, pages 35–40. AAAI Press, 2000.

[Lu97] Luke, S.; Spector, L.; Rager, D. and Hendler J.: Ontology-based Web Agents. In Proceedings of First International Conference on Autonomous Agents, 1997.

[Ol99] Oliver D. E., Shahar Y., Musen M. A. and Shortliffe E. H.: Representation of change in controlled medical terminologies. Artificial Intelligence in Medicine,15(1):53–76, 1999.

[PP99] Martin P. and Eklund P.: Embedding Knowledge inWeb Documents. Proc. 8th Int. World Wide Web Conf. (WWW'8), Toronto, 1999, pp. 1403–1419.

[Mc01] McElroy, M.: Second-Generation Knowledge Management, Macroinnovation Associates, KMWorld 2001, November 2001, Santa Clara, Ca

[OL98] O'Leary, D.E.: Knowledge Management Systems: Converting and Connecting, IEEE Intelligent Systems, 13, May/June, p.30-33.

[Ro96] Roddick, J.F.: A Survey of Schema Versioning Issues for Database Systems, Information and Software Technology, 37(7):383-393, 1996.

[SM01] Staab, S. and Maedche, A.: Knowledge Portals - Ontologies at Work. AI Magazine, 21(2), Summer 2001.

[St01] Staab, S., H.-P. Schnurr, R. Studer, Y. Sure: Knowledge Processes and Ontologies. IEEE Intelligent Systems. 16(1), January/February 2001. Special Issue on Knowledge Management.

[SS00] Staab, S.; Schurr, H.P.: Smart task support through proactive access to organizational memory. Knowledge Based Systems 13(5): 251-260, 2000

[SSV02] Stojanovic, L.; Stojanovic N.; Volz R.: Migrating data-intensive Web Sites into the Semantic Web. To appear: ACM Symposium on Applied Computing SAC, 2002.

[TB01] Tamma V.A.M. and Bench-Capon T.J.M; A conceptual model to facilitate knowledge sharing in multi-agent systems. In Proc. of the OAS 2001. Montreal, May, 29, 2001, pp. 69-76.